

**REMARKS**

Applicants have considered the Office Action mailed on May 7, 2003, and the references cited therewith. This amendment cancels claims 1-12 and 26-39 without prejudice, amends claims 13, 18, and 24, and adds new claims 40-67; as a result, claims 13-25 and 40-67 remain pending in this Application.

Claims 13-25 are drawn to the control-graph augmentation aspect of the present hierarchical-profiling invention. These claims were again rejected as anticipated by or unpatentable over Muchnick, ADVANCED COMPILER DESIGN IMPLEMENTATION (Morgan Kaufmann, 1997), pp. 169-177. The central concept of the invention involves profiling a program as a hierarchical set of regions nested within each other. Muchnick, in contrast, profiles his programs (“procedures”) only as a single entity; none of his blocks (“regions”) are organized hierarchically. Absent guidance from the present Application, one skilled in the art would not even consider Muchnick for this purpose. There are many differences between Applicant’s methods and Muchnick’s.

Independent claim 13, rejected under 35 USC §102(b), has been amended to make explicit what was already implied in its previous form. Specifically, the inner region has “multiple” entry nodes, without which the following “selecting” operation would be meaningless. This operation now selects “a single one” of the “existing” entry nodes as the representative node that stands in for all other nodes in the inner region. “Adding” in the claim operates upon each prolog node having an edge to one of the “existing” entry nodes “other than the representative node.” Edges that already proceed to the representative node would not be “added.”

Page 2, par. 2 of the Office Action states that Muchnick “page 169, first paragraph, global indicates composing hierarchy of flow graphs as presented on page 177, Figure 7.8” This is not correct. First, Muchnick merely desires “a global ‘understanding’” (page 169) of a program; this does not imply that he organizes his programs hierarchically with inner and outer regions. Second, his Figure 7.8 does not show a “hierarchy” of flow graphs. Each of the dotted boxes in that figure encloses a flow graph that is separate from and outside each of the other

boxed flow graphs. That is, the “inner region” and “outer region” of claim 13 are not found in Muchnick.

Claim 13 asserts “multiple entry nodes” for the inner block. First, Muchnick has no “inner” regions, as discussed above. Second, Muchnick mentions multiple “initial blocks” in the footnote to page 174. The Office Action equates Muchnick’s “blocks” with Applicant’s “regions” in the passage quoted in the above paragraph, but then inconsistently equates them with Applicant’s “nodes” for this purpose. Although claim 13 recites “replacing” a region with a node, they are not the same thing.

Also, Muchnick never replaces anything. For example, the dotted boxes in Figure 7.8 merely identify blocks. The blocks inside the boxes, for example, do not take the place of the box, as claim 13’s representative entry node “replac[es]” the inner region.

Claim 13 recites “a prolog node” to an entry node. The Specification defines this term on page 8 lines 16-18. An essential characteristic of a prolog node is that it exists at a different level in the hierarchy of the program. Muchnick has no hierarchy, and thus has no prolog nodes. In like manner, Muchnick has no “epilog node,” as that term is defined on page 8 lines 27-29.

Prolog node or no, Muchnick further does not show or suggest “adding” an edge as defined in claim 13. Muchnick adds a single entry node and a single exit node to the entire program, as described in the last paragraph of page 174 and shown in Figure 7.8; these nodes have no effect other than to provide convenient points for describing the beginning and end of the whole program. In any case, they are not “existing” entry nodes which claim 13 selects, nor are they nodes “other than the representative entry node” in the inner region to which claim 13 adds an edge.

Dependent claims 14-17 incorporate all the features of parent claim 13. The secondary reference to Ball and Adl-Tabatabai, cited in rejecting claims 14 and 17 under 35 USC §103(a), do not supply any of the deficiencies of the primary Muchnick treatise.

Independent claim 18 similarly recites “inner” and “outer” regions, not found in the non-hierarchical teachings of Muchnick. The “multiple” inner-block entry nodes, and the “single” “selected” entry node distinguish Muchnick for the same reasons as described in connection with

claim 13. Claim 18 also includes the “prolog” and “epilog” nodes absent from the reference, and recites “adding” edges therefrom to “existing” entry and exit nodes.

Dependent claims 19-23 incorporate all the features of parent claim 13. The secondary references to Adl-Tabatabai and Ball, cited in rejecting claims 22 and 23 under 35 USC §103(a), do not make up any of the deficiencies of the primary reference.

Article claim 25 incorporates all the method elements of claim 18, and therefore also distinguishes Muchnick.

Claims 1-12 and 26-39 were rejected as unpatentable over Ball, “Efficient Path Profiling” (1996), pp. 46-57, in view of U.S. patent 6,170,083 to Adl-Tabatabai. Applicant again reserves the right to antedate this latter reference. These claims are canceled without prejudice herein, replaced with new claims 40-67.

New claims 40-49 concern region-based profiling. Independent claim 40 specifies that at least some of the “hierarchical” regions have “multiple entry nodes and/or multiple exit nodes.” Ball has no hierarchical regions, and neither Ball nor Adl-Tabatabai teaches regions with multiple entry or exit nodes. Their techniques would not operate if they should define regions having more than a single entry or a more than one exit. If the selected region has multiple entry nodes—which Ball and Adl-Tabatabai cannot—claim 49 adds an edge to “all the entry nodes.” In like manner, if the selected region has multiple exit nodes—not permitted in Ball and Adl-Tabatabai—claim 49 adds an edge from “all the exit nodes.” Claim 40 then assigns unique-sum identifiers to these edges, “including the added edges.” Neither reference teaches this operation, nor suggests any reason for performing it.

Claims 41-48 depend from claim 40. They distinguish the references for the same reasons, and for others as well. For example, claim 46 initializes the combination (e.g., sum) to different identifiers (e.g., numbers), “depending upon which of the multiple entry nodes is executed during the single execution,” as shown at node R<sub>e</sub> in Fig. 3A of the Application. The references always initialize to the same number upon entering a block. As another example, claim 48 specifies “has multiple exit nodes, only one of which is executed during a single execution of the region.” The references have only single exit blocks.

Claim 49 includes all the method operations of claim 40, and thus also distinguishes the references.

Claims 50-67 relate to global-based profiling, as shown in Figs. 7-10 of the Application. Independent claim 50 inserts additional edges into multiple regions of the program, at least some of these edges “extending between an outer one of the regions and an inner one of the regions,” such as edges between nodes A and C in Fig. 8A, and between D-G, D-I, F-I, and G-I. Neither Ball nor Adl-Tabatabai add any such nodes to a CFG of a program. Claim 50 further assigns identifiers (e.g., numbers) to the existing edges, and also to “at least some of the additional edges,” such as the aforementioned edges in Fig. 8A. Neither cited reference assigns numbers to edges added to a CFG. In the references, path sums are unique only through paths involving the originally existing edges.

Claims 51-66 depend from claim 50, and adduce other features as well. For example, claim 53 selects “a representative path,” whereas the references treat all possible paths equally. Claims 55-62 set forth the rules for adding specific edges as summarized on page 17 line 30 to page 18 line 20 of the Application. The references add no such edges. Neither reference suggests blocks or regions having multiple entry or exit nodes as recited in claims 58 or 61.

Dependent claims 64-66 concern the execution of a CFG augmented for global-based profiling as in claim 50. Claim 66, for example, specifies that “the number of counters is less than the possible number of plain paths through the program,” as described on page 20 lines 2-4. Neither Ball nor Adl-Tabatabai teaches such sparse arrays of counters.

Claim 49 distinguishes the references by including all the method operations of claim 50.

**Conclusion**

Applicants urge that the claims are in condition for allowance and respectfully request reexamination and notification to that effect. The Examiner is invited to telephone Applicants' attorney at 612-373-6971 to facilitate prosecution of this Application.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 19-0743.

Respectfully submitted,

YOUFENG WU ET AL.

By their Representatives,

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.  
Attorneys for Intel Corporation  
P.O. Box 2938  
Minneapolis, Minnesota 55402  
612-373-6971

Date 25 JUNE 2003

By J. Michael Anglin  
J. Michael Anglin  
Reg. No. 24,916

**CERTIFICATE UNDER 37 CFR 1.8:** The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Mail Stop AF, Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on this 25 day of June, 2003

KACIA LEE  
Name

Kacia Lee  
Signature